

PROCESADORES AVANZADOS. PIC 16F877.  
EPSG-UPV.  
PROGRAMA REALIZADO PARA EL COMPILADOR HITECH.

# SISTEMA DE CONTROL DE TEMPERATURA

JOSE LUIS OLCINA LLOPIS.  
[josele.ono@hotmail.com](mailto:josele.ono@hotmail.com)  
Benilloba.

## SISTEMA DE CONTROL DE TEMPERATURA DE UN CLIMATIZADOR:

Para esta aplicación se utilizan 2 microinterruptores de 8 bits, una placa con el sensor de temperatura LM35, un display LCD 16x2 y una placa con diodos led.

Los microinterruptores los conectaremos al PORTC y al PORTD. El bus de datos del LCD al PORTB y los bits de control E W S a los pines RE0 RE1 RE2 respectivamente. El LM35 a la patilla RA0 y dos de los leds de la placa de diodos a las patillas RA1 y RA2.

### FUNCIONAMIENTO DEL SISTEMA:

El usuario introduce la temperatura máxima y mínima deseadas mediante los microinterruptores conectados al PORTD y PORTC respectivamente.

Si el sistema supera la temperatura máxima del rango, se activara una alarma visual con el texto “FRIO” en el LCD y se apagara\* el led conectado a la patilla RA2.

ACTUAL:28	FRIO
MAX:27	MIN:24

Cuando el sistema no alcance la temperatura mínima del rango, se activará la alarma visual “CALOR” en el LCD y se producirá un cambio de nivel de tensión en la patilla RA1 apagándose\* el led conectado a esta.

ACTUAL:23	CALOR
MAX:27	MIN:24

\*Los leds se apagaran debido a que por defecto estarán encendidos al alimentar la placa de leds utilizada. De esta forma podremos visualizar mejor los cambios de nivel.

Si por algún casual, el usuario introduce un valor de temperatura mínima mayor que el valor de temperatura máxima, se activará una alarma visual con el texto “ERROR”, y por supuesto no se producirá ningún cambio de nivel en RA1 ni en RA2.

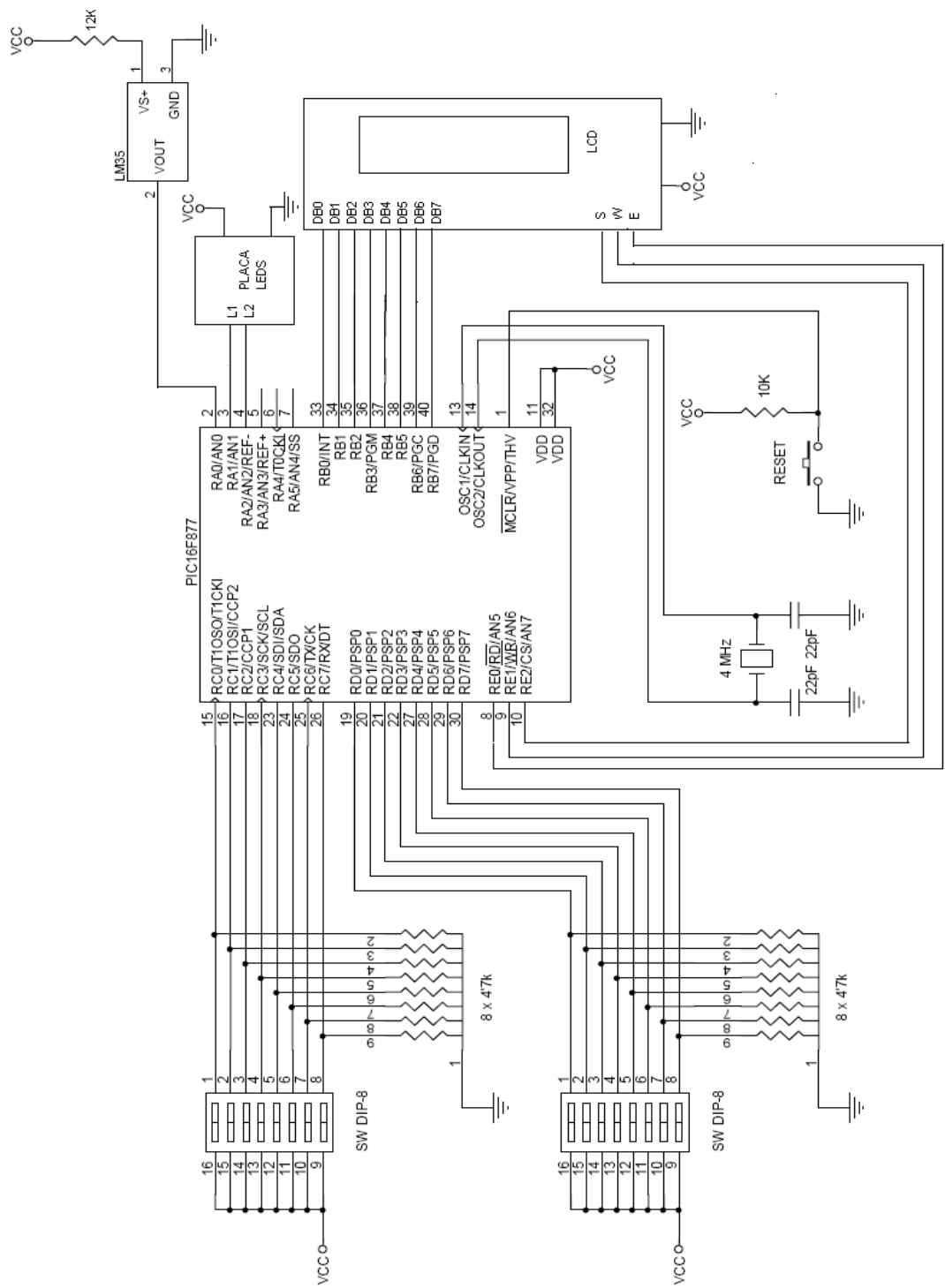
ACTUAL:28	ERROR
MAX:22	MIN:24

Siempre que el usuario introduzca correctamente las temperaturas máxima y mínima y la temperatura ambiente permanezca dentro del intervalo impuesto, visualizaremos únicamente en el LCD la temperatura actual y la mínima y máxima elegidas.

ACTUAL:26	
MAX:27	MIN:24

En las líneas 2 y 3 de la página 6 podemos activar la función prueba2( ) y desactivar rango( ), de esta manera veríamos en la segunda línea del LCD la temperatura máxima y mínima alcanzadas por el sensor.

ESQUEMA DEL CIRCUITO:



## PROGRAMA EN C:

```
//PROGRAMA DE UN SISTEMA Q MIDE LA TEMPERATURA ACTUAL,
//ADEMÁS MIDE LA MAXIMA Y MINIMA (O LIMITA LA TEMPERATURA A
//UN RANGO IMPUESTO POR EL USUARIO).
// SI EL SISTEMA SUPERA LA TEMPERATURA MAXIMA DEL RANGO, SE
//ACTIVARA LA ALARMA "FRIO" Y SE APAGARA EL LED CONECTADO A LA
//PATILLA RA2.
//SI POR EL CONTRARIO, EL SISTEMA NO SUPERA LA TEMPERATURA
//MINIMA DEL RANGO, SE ACTIVARA LA ALARMA "CALOR" Y SE
//APAGARA EL LED CONECTADO A LA PATILLA RA1.
//LOS LEDS SE APAGARAN DEBIDO A QUE POR DEFECTO LOS LEDS
//ESTARAN ENCENDIDOS AL ALIMENTAR LA PLACA DE LEDS

//EN PORTC SE INDICARA LA TEMPERATURA MINIMA DEL RANGO
//MEDIANTE MICROINTERRUPTORES.
//EN PORTD SE INDICARA LA TEMPERATURA MAXIMA DEL RANGO
//MEDIANTE MICROINTERRUPTORES.
//SI PORTC>PORTD VISUALIZAREMOS LA ALARMA "ERROR".

//*****
//**CABECERA Y DECLARACION DE VARIABLES Y FUNCIONES*****

#include<pic1687x.h>

unsigned char x=1; //para el switch (inicializa y actualiza max y min)
unsigned int temp,tempmin,tempmax,t; //variables en hexadec y t (retardo conversor).
unsigned int templcd,tempminlcd,tempmaxlcd; //variables en decimal para enviar al lcd.
unsigned int rangmin,rangmax; //variables en Hexadecimal para enviar al lcd.
//de los interruptores.

unsigned char hexadec (unsigned char hex); //Pasa a decimal.
void iniciar_lcd(void); //Inicia el LCD.
void lcd_control(unsigned int valor); //Control del LCD (Instrucciones).
void prueba(void); //Texto y datos a enviar al LCD.
void prueba2(void); //Texto y datos a enviar al LCD.
void rango(void); //Texto y datos a enviar al LCD.
void prueba3(void); //Texto y datos a enviar al LCD.
void prueba4(void); //Texto y datos a enviar al LCD.
void error(void); //Texto y datos a enviar al LCD.
void lcd_dato(unsigned int dato); //Envia al LCD.

unsigned int i; //para el retardo de las funciones.
```

```

//*****
//*****MAIN*****
//*****

void main(void){

    TRISB=0x00;           //SALIDA
    TRISE=0x00;           //SALIDA

    TRISC=0xFF;           //ENTRADA DIGITAL
    TRISD=0xFF;           //ENTRADA DIGITAL
    TRISA=0x01;           //salida digital. RA0 IN ANALOG

    PORTA=0x00;           //PORTA SALIDAS DIGITALES A 0
    PORTC=0x00;           //TEMP MINIMA DEL RANGO
    PORTD=0x00;           //TEMP MÁXIMA DEL RANGO

    PORTE=0x00;           //INICIALIZO A 0
    PORTB=0x00;

    OPTION=0x80;
    INTCON=0xC0;
    ADCON1=0x8E;           //justificado a derecha
    ADCON0=0x81;

    templcd=tempminlcd=tempmaxlcd=37;    //37dec=25hex temperatura por defecto

    ADIE=1;                //HABILITO INTERRUPCION.
    ADGO=1;                //PONGO EN MARCHA LA CONVERSIÓN.

    while(1);

} //MAIN

//*****FIN DEL MAIN*****
//*****

```

```
//*****INTERRUPCION CONVERSION A/D*****
```

```
void interrupt convertir (void){
```

```
if(ADIF){                                     //SI HA TERMINADO LA CONVERSION
    ADIF=0;
    temp=(ADRESH<<8)+ADRESL;    //JUSTIFICADO A DERECHA EN ADCON1.

    temp=temp>>1;                //divido entre 2 para buscar paridad de la escala.
    templcd=hexadec(temp);       //PASO A DECIMAL.

    rangmin=PORTC;               //temp mínima introducida interruptores.
    rangmax=PORTD;               //temp mínima introducida interruptores.

    iniciar_lcd();
    lcd_control(0x80);           //fija mensaje en la primera fila.
    prueba();                   // envia al lcd.
```

```
//Si queremos visualizar la temperatura minima y maxima q alcanza el termometro:
```

```
//*****Inicializa y Actualiza la temperatura mínima y máxima*****
```

```
//La idea es q si la primera vez q conectamos el sistema y:
```

```
// 1)la temp actual(28 por ej) es mayor a la minima (25 por defecto),la minima sea 28.
// 2)la temp actual (23 por ej) es menor a la maxima (25 por defecto), la maxima sea 23.
```

```
switch(x){
case 1:
    if(templcd>tempminlcd) {                //podra entrar solo una vez
        tempminlcd=templcd;
        }x++;break;
case 2:
    if(templcd<tempmaxlcd) {                //podra entrar solo una vez
        tempmaxlcd=templcd;
        }x++;break;
case 3:
    if (templcd<tempminlcd) {
        tempminlcd=templcd;
        }x++;break;
case 4:
    if(templcd>tempmaxlcd) {
        tempmaxlcd=templcd;
        }x=3;break;

} //switch
//*****
```

```

lcd_control(0xC0);      //MENSAJE EN 2ª FILA
    //prueba2();        //prueba2() es para visualizar la max y min del termometro
    rango();            //rango() es para visualizar la max(PORTD)y la min(PORTC)
                        //introducidas por el usuario

```

```

//CONDICIONES PARA ACTIVAR LA ALARMA Y APAGAR LOS LEDS RA1 Y RA2.
//EN UN PRINCIPIO LOS LEDS ESTAN TODOS ON (CONDICION DE LA PLACA).

```

```

if (templcd < PORTC){    //SI LA TEMP ACTUAL ES MENOR Q LA MINIMA Q INTRODUCIMOS

    if (PORTC>PORTD){    //SI LA TEMP MIN RANGO>TEMP MAX RANGO=>ERROR.

        lcd_control(0x80); //primera fila del lcd
        error();           //rango mal introducido
        RA1=1;            //desactivamos led calor, ya q hay error en el rango
    }

    else{                //SI NO...

        lcd_control(0x80); //primera fila del lcd
        RA1=0;            //APAGAMOS LED RA1
        prueba3();        //MOSTRAMOS CALOR EN LCD
    }

}

```

```

if (templcd > PORTD){    //SI LA TEMP ES MAYOR Q LA Q LE INTRODUCIMOS
                        //(LA MAXIMA Q QUEREMOS).

    if (PORTC>PORTD){    //SI LA TEMP MIN RANGO>TEMP MAX RANGO=>ERROR.

        lcd_control(0x80); //primera fila del lcd
        error();           //rango mal introducido
        RA2=1;            //desactivamos led frio, ya q hay error.
    }

    else{                //SI NO...

        lcd_control(0x80); //primera fila del lcd
        RA2=0;            //APAGAMOS LED RA2
        prueba4();        //MOSTRAMOS FRIO EN LCD
    }

}

```

```

//*****PARA VOLVER A ENCENDER LOS LEDS RA1 RA2*****

if (templcd<=PORTD){      //SI LA TEMP <= A LA MAXIMA INTRODUCIDA
    RA2=1;                //ENCENDEMOS LED FRIO.
}

if (templcd>=PORTC){      //SI LA TEMP>= A LA MINIMA INTRODUCIDA
    RA1=1;                //ENCENDEMOS LED CALOR..
}

//*****

for (t=0;t<50000;t++);    //retardo antes de volver a activar conversor

    ADGO=1;

        }//IF de if (ADIF)

} //INTERRUPT

//*****FIN INTERRUPCION*****

```

```

//*****
//*****ZONA DE FUNCIONES*****

```

```

unsigned char hexadec (unsigned char hex){

```

```

    unsigned char x,y;

```

```

    x=hex/0x0A;

```

```

    x=x<<4;

```

```

    y=hex%0x0A;

```

```

    return (x+y);

```

```

} //unsigned char hexADEC

```

```

void iniciar_lcd(void){

```

```

    PORTB=0x0E;

```

```

    //INICIO EL LCD(texto visible,cursor vis,no parpadea)

```

```

    RE0=1;

```

```

    //ACTIVO ENABLE

```

```

    for(i=0;i<=100;i++);

```

```

    //aplico retardo

```

```

    RE0=0;

```

```

    //DESACTIVO ENABLE

```

```

    for(i=0;i<=100;i++);

```

```

    //aplico retardo

```

```

    PORTB=0x3C;

```

```

    //Juego de datos(bus 8 bits,2 lineas,5x10 puntos)

```

```

    RE0=1;

```

```

    //ENABLE

```

```

    for(i=0;i<=100;i++);

```

```

    //aplico retardo

```

```

    RE0=0;

```

```

    //ENABLE

```

```

    for(i=0;i<=100;i++);

```

```

    //aplico retardo

```

```

return;

```

```

}

```

```

void lcd_control(unsigned int valor){

```

```

    RE2=0;

```

```

    //INDICA Q ES INSTRUCCION (RS=0)

```

```

    for(i=0;i<=100;i++);

```

```

    //aplico retardo

```

```

    PORTB=valor;

```

```

    //PORTB=0x80 => primera linea

```

```

    for(i=0;i<=100;i++);

```

```

    //aplico retardo

```

```

    RE0=1;

```

```

    //ENABLE

```

```

    for(i=0;i<=100;i++);

```

```

    //aplico retardo

```

```

    RE0=0;

```

```

    //ENABLE

```

```

    for(i=0;i<=100;i++);

```

```

    //aplico retardo

```

```

return;

```

```

}

```

```

void prueba(void){          //TEXTO y TEMPERATURA A ENVIAR AL LCD.

lcd_dato('A');
lcd_dato('C');
lcd_dato('T');
lcd_dato('U');
lcd_dato('A');
lcd_dato('L');
lcd_dato(':');
lcd_dato( 0x30 | (templcd & (0xF0))>>4 ); //D7-D4=0011. D3-D0=templcd&(0xF0);
lcd_dato( 0x30 | ( templcd & (0x0F) ) );    //D7-D4=0011. D3-D0=templcd&(0x0F);
lcd_dato(' ');
lcd_dato(' ');
lcd_dato(' ');
lcd_dato(' ');
lcd_dato(' ');
lcd_dato(' ');
lcd_dato(' ');
lcd_dato(' ');
}

```

```

void prueba2(void){        //TEXTO Y TEMPERATURA A ENVIAR AL LCD

lcd_dato('M');
lcd_dato('A');
lcd_dato('X');
lcd_dato(':');
lcd_dato(0x30 | (tempmaxlcd & (0xF0))>>4 ); //D7-D4=0011. D3-D0=tempmax&(0xF0);
lcd_dato(0x30 | ( tempmaxlcd & (0x0F) ) );    //D7-D4=0011. D3-D0=tempmax&(0x0F);
lcd_dato(' ');
lcd_dato(' ');
lcd_dato(' ');
lcd_dato(' ');
lcd_dato('M');
lcd_dato('T');
lcd_dato('N');
lcd_dato(':');
lcd_dato(0x30 | (tempminlcd & (0xF0))>>4 ); //D7-D4=0011. D3-D0=tempmin&(0xF0);
lcd_dato(0x30 | ( tempminlcd & (0x0F) ) );    //D7-D4=0011. D3-D0=tempmin&(0x0F);
lcd_dato(' ');

}

```

```

void rango(void){          //TEXTO Y TEMPERATURA A ENVIAR AL LCD

lcd_dato('M');
lcd_dato('A');
lcd_dato('X');
lcd_dato('.');
lcd_dato(0x30 | (rangmax & (0xF0))>>4 ); //D7-D4=0011. D3-D0=tempmax&(0xF0);
lcd_dato(0x30 | ( rangmax & (0x0F) ) );    //D7-D4=0011. D3-D0=tempmax&(0x0F);
lcd_dato(' ');
lcd_dato(' ');
lcd_dato(' ');
lcd_dato(' ');
lcd_dato('M');
lcd_dato('I');
lcd_dato('N');
lcd_dato('.');
lcd_dato(0x30 | ( (rangmin & (0xF0))>>4 ) ); //D7-D4=0011. D3-D0=rangmin&(0xF0);
lcd_dato(0x30 | ( rangmin & (0x0F) ) );      //D7-D4=0011. D3-D0=rangmin&(0x0F);
lcd_dato(' ');

}

```

```

void prueba3(void){      //TEXTO Y TEMPERATURA A ENVIAR AL LCD.

lcd_dato('A');
lcd_dato('C');
lcd_dato('T');
lcd_dato('U');
lcd_dato('A');
lcd_dato('L');
lcd_dato('.');
lcd_dato( 0x30 | (templcd & (0xF0))>>4 ); //D7-D4=0011. D3-D0=templcd&(0xF0);
lcd_dato( 0x30 | ( templcd & (0x0F) ) );    //D7-D4=0011. D3-D0=templcd&(0x0F);
lcd_dato(' ');
lcd_dato('C');
lcd_dato('A');
lcd_dato('L');
lcd_dato('O');
lcd_dato('R');
lcd_dato(' ');
lcd_dato(' ');

}

```

```

void prueba4(void){          //TEXTO Y TEMPERATURA A ENVIAR AL LCD.

    lcd_dato('A');
    lcd_dato('C');
    lcd_dato('T');
    lcd_dato('U');
    lcd_dato('A');
    lcd_dato('L');
    lcd_dato(':');
    lcd_dato( 0x30 | (templcd & (0xF0))>>4 ); //D7-D4=0011. D3-D0=templcd&(0xF0);
    lcd_dato( 0x30 | ( templcd & (0x0F) ) );    //D7-D4=0011. D3-D0=templcd&(0x0F);
    lcd_dato(' ');
    lcd_dato('F');
    lcd_dato('R');
    lcd_dato('I');
    lcd_dato('O');
    lcd_dato(' ');
    lcd_dato(' ');
    lcd_dato(' ');

}

void error(void) {

    lcd_dato('A');
    lcd_dato('C');
    lcd_dato('T');
    lcd_dato('U');
    lcd_dato('A');
    lcd_dato('L');
    lcd_dato(':');
    lcd_dato( 0x30 | (templcd & (0xF0))>>4 ); //D7-D4=0011. D3-D0=templcd&(0xF0);
    lcd_dato( 0x30 | ( templcd & (0x0F) ) );    //D7-D4=0011. D3-D0=templcd&(0x0F);
    lcd_dato(' ');
    lcd_dato('E');
    lcd_dato('R');
    lcd_dato('R');
    lcd_dato('O');
    lcd_dato('R');
    lcd_dato(' ');
    lcd_dato(' ');

}

```

```

void lcd_dato(unsigned int dato){           //ENVIA AL LCD.

    RE2=1;                                //RS=1; lee dato
    for(i=0;i<=100;i++);                  //aplico retardo
    PORTB=dato;
    for(i=0;i<=100;i++);                  //aplico retardo
    RE0=1;                                //ENABLE
    for(i=0;i<=100;i++);                  //aplico retardo
    RE0=0;                                //ENABLE
    for(i=0;i<=100;i++);                  //aplico retardo
    return;

}
//*****FIN DE FUNCIONES*****

```